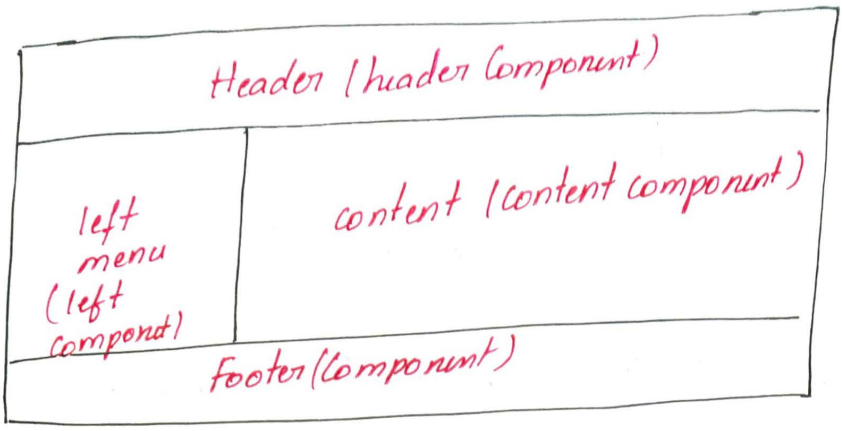
Component :-



- Component is a basic building block of angular
- The component encapsulates the data, html and a logic for a view
- You can create n number of component in the project.
- Component in other word is a simple typescript class which contains template, style & typescript class.
- The component contains meta-data like template url, styleUrl & animation effect.
- Creating component through terminal

ng g c student

↓

component name.

ng stands for Angular

g stands for generate

c stands for Component

Whenever we create a component through terminal it creates 4 files

1. student.component.css 2. student.component.html
3. student.component.spec.ts 4. student.component.ts
5. updates app.module.ts.

Whenever the component is created it will be registered in app.module.ts in the declarations

```
declarations:
[
  AppComponent,
  StudentComponent
]
```

along with that the student component is imported by dufault.

```
import { StudentComponent } from './student/
                    student.component';
```

index.html

```
<html>
  <head> </head>
  <body>
    <app-root>  </app-root>  // app.component will be activated.
  </body>
</html>
```

AppComponent.ts

```
@Component({
selector: 'app-root'
templateUrl: './app.component.html',
styleUrls: ['./app.componed.css']
})
export class AppComponent{
    title = 'App1';
}
```

app.component.html

```
<h1> Welcome to Angular
                </h1>
<h5> We are in app-comp.
                </h5>
<app-student></app-
                student>
<h1> end of app
  component </h1>
```

app.component.css

```
h1
{
  color: red;
}
```

student.component.ts

```
import { Component, OnInit } from '@angular/core';
@Component({
    selector: 'app-student',
    templateUrl: './student.component.html',
    styleUrls: ['./student.component.css']
})
export class StudentComponent implements OnInit
{
    constructor()
    {
    }
    ngOnInit(): void { }
}
```

student.component.html

```
<p> student works! </p>
```

Output:-

Welcome to first angular app

Good Morning

We are in app.component.html

student works! → student component.

end of the app component

Example:- External Styling.

index.html

```
<body>
  <app-root></approot>
</body>
```

app.component.html

```
<h1> AppComponent</h1>
<app-student> </app-student>
```

app.component.css

```
h1{
   color: red
}
```

student.component.html

```
<h1>We are inserting in
student component.html</h1>
<p> Student works</p>
```

student.component.css

```
h1
{
  colors green
}
```

Output:-

app component

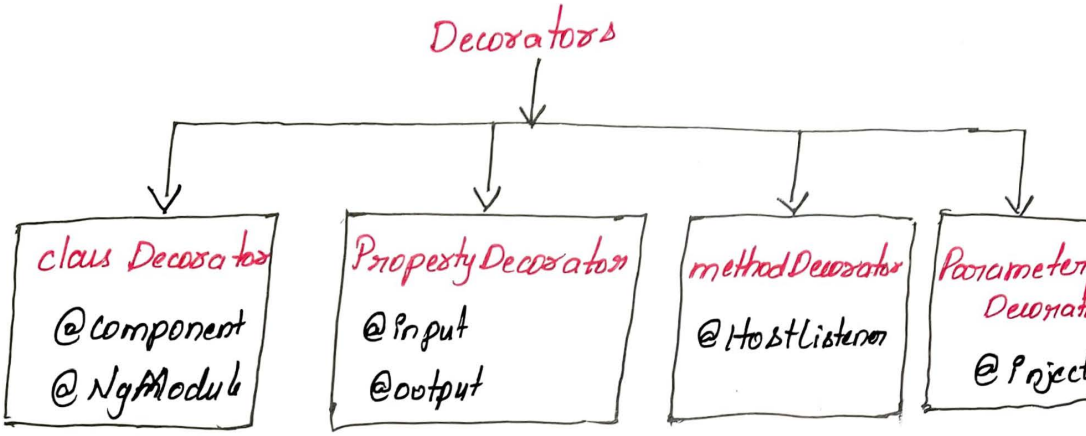
text <- colors green we are inserting in student component.html

student works!

Decorators:-

- Decorators are feature of typescript and are implemented as functions.
- Decorator starts with @
- Decorators are simple functions. These functions supply meta-data to angular (class, property, method)
- Decorators are invoked at run-time
- It allows you to execute functions.

Types of Decorators



Claus Decoration:

- @NgModule({}) is present in module.ts
- @Component({}) is present in component.ts
- NgModule ex: app.module.ts
- Component ex: app.component.ts

Propesty Decorator:

@input.
@output.

Method Decoration:

@HostListener:- It is a method decoration which listens to the response of the host.

app.component.ts

```
@HostListener('click', ['$event'])
onhostclick(event: Event)
{
  alert("Good Afternoon")
  consol.log(event)
}
```

O/p:-

On Consol screen:-

MouseEvent {isTrusted: true, screenX: 312 screenY: 191}

Execution related to typscript file i.e., module and component

index.html

```
<body>
<app-root></app-root>
</body>
```

app.module.ts

```
export class AppModule
{
  constructor()
  {
    console.log("we are from module.ts file")
  }
}
```

app.component.ts

```
import {Component} from '@angular/core';
@Component({
  - - - -
  - - - -
})
export class AppComponent
{
  constructor()
  {
    console.log("we are from component.ts file")
  }
}
```

Output:-

Browser Screen

App Component.

Console screen

hello we are in module ts file

hello we are in component ts file

Parameter Decorator:

- We have to create a service the command is

```
ng g service myser
```

- It will be creating two files

```
CREATE src/app/myser.service.spec.ts
CREATE src/app/myser.servic.ts
```

- whenever you create a service it has to be registered in the providers. Once it is registerd in the providers it has to be imported.